

Eliminating ditransitives

András Kornai

Harvard University Institute for Quantitative Social Science
and
Computer and Automation Research Institute, Hungarian Academy of Sciences
andras@kornai.com

Abstract. We discuss how higher arity verbs such as *give* or *promise* can be treated in an algebraic framework that admits only unary and binary relations and does not rely on event variables.

Introduction

Until the groundbreaking work of Russell (1900), ideas of semantic representation centered on the Aristotelian notion that the predicate inheres in (is attributed to) the subject. In modern terminology, this amounts to admitting only unary relations such as `dog(x)` or `barks(y)` and treating binary relations such as `MARRY(X,Y)` as the conjunction of unaries `marry(x) & marry(y)`. (For greater clarity, unaries will be given in `typewriter` and binaries in `SMALL CAPS` font.) As Russell pointed out, such an analysis will of necessity treat all binary relations as symmetrical, with intolerable consequences for those relations that are asymmetrical such as `GREATER_THAN(X,Y)` or `FATHER_OF(X,Y)`. A Davidsonian analysis trivially eliminates ditransitives and higher arity verbs, but only at the price of introducing an event variable, a step of dubious utility for statives like *has*. We follow Russell in admitting at least one asymmetric relation, which we will denote ‘<’, and perhaps a handful of others such as `HAS(x,y)` ‘x possesses y’ `AT(x,y)` ‘x is at location y’, `CAUSE(x,y)`, etc.

While we are obviously not disputing Russell’s key observation, we believe the remedy he proposed was far too radical, throwing out all the linguistic insight that comes with the subject/predicate analysis. In this paper we propose to retrench, both in terms of drastically reducing the number of available binary relations and in terms of eliminating ternary and higher order relations entirely. To illustrate the main ideas in Section 1 we begin with a typical higher arity verb, *promise*, which is generally treated as involving at least three, but possibly as many as five, open slots: an agent, the promisor; a recipient, to whom the promise is made; the object of the promise; and perhaps an issue date and a term date as in *Alice promised Carol on Monday that she will get her twenty bucks back before Friday*. In Section 2 we present the tectogrammar, which has its roots in the decomposition technique long familiar from generative semantics (Lakoff 1968), whereby *kill* is analyzed as ‘cause to die’ and *give* as ‘cause to have’ – we discuss what makes the current model immune to Fodor’s (1970)

critique. In Section 3 we present the formal model using a classic generalization of finite state automata (FSA) and finite state transducers (FST), *machines* (Eilenberg 1974). In the concluding Section 4 we discuss how this approach differs both from the standard model-theoretic approach and the less standard, but widely used systems of knowledge representation by semantic networks such as presented in Quillian (1969), Brachman (1979), or Sowa (2000), which retain a fundamentally Aristotelian character. We argue that the elimination of ditransitives makes possible a fundamental simplification in the network mechanism in that we no longer need to deal with hypergraphs where ‘edges’ could be node sets of arbitrary size – ordinary graphs will suffice.

1 The semantics of *promise*

What does it mean to make, and keep promises? As Rawls (1955:16) puts it, “The point of the practice is to abdicate one’s title to act in accordance with utilitarian and prudential considerations in order that the future may be tied down and plans coordinated in advance”. Our goal is not to dispute what Rawls says, indeed we take this to be a perfectly reasonable explanation of why the social practice of promise keeping is useful, our goal here is simply to explicate all the hidden implicational background assumed by Rawls and by users of English in general.

A *promise* is a commitment to some future action or some state of affairs that can be brought about by such action. It is assumed that the promissor is someone who can either perform the action in an agentive fashion, or that the promise pertains to the actions of someone or something under the control of the promissor. Thus *I will have the car ready by 8AM tomorrow* or *No, he won’t make a mess* are well-formed promises, while *Water boils at 100 degrees centigrade I promise* is dubious usage, and *You will win the lottery/I will cure your cancer* are suspect on their face. To make an explicit promise encompasses an implicit statement by the promissor that they be capable of either performing the action themselves, or be capable of inducing someone/something to perform it for them. We will not have much to say about those cases, such as promising the boiling point of water, that can be paraphrased as ‘I’m informing you’, beyond the simple observation that this pertains to the knowledge state of the promissor, and in fact the promissor would be the first to admit this. But we are crucially interested in cases such as *I will cure your cancer* or *I promise eternal life* where the ability of the promissor to deliver is in grave doubt.

Let X be a predicate of some sort, and let $P(A, X, T_0, T)$ be the statement ‘at time T_0 A promises X will hold at time T ’. We need at least a concept of linear order of time (since *I promise you won’t have to wear a scarf tomorrow* is meaningful in a way that *I promise you didn’t have to wear a scarf yesterday* is not) and the condition $T_0 < T$. Further, we need a notion of agency that restricts the overall set of promises to keepable ones, thus distinguishing *I promise I will bring the book tomorrow* from *I promise I will win the lottery tomorrow*. Broadly speaking, there are actions (or states of affairs – from here on we will just speak of

‘matters’) that are within our power, and there are matters that are not: fetching a physical object generally falls in the first category, suddenly becoming wealthy falls in the second. We need a predicate $C(A, X, T)$ which means ‘agent A can control matter X at time T ’. Such control can be physical, as in the case of bringing the book, or purely notional, as in the case of a judge declaring some contract null and void. It is here that the emptiness of the promise about the boiling point of water becomes evident: clearly, whatever this boiling point is (actually, it is 99.97 °C at normal atmospheric pressure), there is no person who can change it.

So far, we have $P(A, X, T_0, T) \Rightarrow C(A, X, T)$ where the \Rightarrow is some sort of normative implication: $U \Rightarrow V$ means that if U is reasonable we can reasonably expect V or, what is the same, if V does not hold U cannot be reasonably expected. Thus, a reasonable person A will not promise that someone will win the lottery because we (any reasonable person, A inclusive) don’t expect that A can control the outcome of the drawing. If A is an employee of the sweepstakes company the expectations are different, and a (criminal) promise can possibly be made, but we’d still want to know a great deal more about the causal chain whereby this control over the drawing (or perhaps over the recording or the announcement of the results) is exerted. Notice that the test of reasonableness is not any different for those cases where our default assumption is the presence, rather than the absence, of control: we assume owners control their dogs and parents control their babies, yet we remain slightly dubious in regards to promises such as *He won’t make a mess* precisely because we don’t necessarily see the promisor as having the requisite degree of control over the matter.

For control, at minimum we need a matter that can be both ways: unless we have $M(X, T)$ and $M(\neg X, T)$ (where M is some possibility operator ‘might’) there cannot be any controller of X . What does $M(X, T)$ mean? Certainly $X(T)$, the fact that X holds at T , is sufficient to guarantee that X *might* hold at T , but it is either the case that $X(T)$ or it is the case that $\neg X(T)$ so knowing the state of the matter X at T is insufficient – this is well-traveled ground in modal logic. If the only possible worlds are the states of the actual world at different time instances, $M(X, T)$ implies $\exists T_1 X(T_1) \wedge \exists T_2 \neg X(T_2)$. If there are different alternatives with different timelines this becomes more complicated, but for our purposes we can get by with the simple view and our simple notion of natural or default implication \Rightarrow . Fortunately, we already have a different time instance at hand, namely the time T_0 when the promise is made. The thesis we will defend here includes the somewhat radical abductive inference that this is all that is required: the whole modal apparatus can be dispensed with in favor of the view that a promise is actually a promise to change, $P(A, X, T_0, T) \Rightarrow (\neg X(T_0) \wedge X(T))$.

At first blush, such a view seems to disallow all promises aimed at keeping some state of affairs intact. Since our goal is to offer a theory of ordinary language use, ignoring canonical cases of promises, such as marital vows, which are rather clearly aimed at preserving a certain state of affairs, is not an option, and we need to discuss how these fit in our model. The key issue, as we shall see, is the

semantics of the modal operator *might*, which, as we will argue, already carries this implication of change. Before turning to this, let us simplify the example a bit. Marital vows are rather complex in that they require the presence of two agents and have an aspect of mutuality, so to simplify matters we use a promise of (continued) non-smoking as our example. We claim that the difference between a promise to quit, *This was my last cigarette*, where $\text{smokes}(T_0)$ is to be followed by $\neg \text{smokes}(T)$ for $T > T_0$, and a promise to stay the course, where the expectation is the exact same $\neg \text{smokes}(T)$ for $T > T_0$, is a matter of accommodation: what is hearer assumes in such cases is that the non-smoking behavior at T_0 (the time of making the promise) was accidental. We make this argument indirectly: suppose that $\neg \text{smokes}(T_0)$ was not accidental, it was already the result of a promise. But renewal of a promise would be an empty gesture, for either the original promise was valid, in which case it remains binding for all future times, or it was not, in which case we cannot reasonably expect the promisor to uphold the renewed promise in light of non-performance on the earlier one. Therefore, by the usual quality implicature, we assume that any promisor is a non-accidental non-smoker for the first time. A general consequence of this line of argument is that it is pragmatically impossible to re-promise something.

Turning to the modal M we see that $\exists T_1 X(T_1) \wedge \exists T_2 \neg X(T_2)$ does not exhaust the meaning of $M(X, T)$. First of all, if this were sufficient, from $\exists T_1 X(T_1) \wedge \exists T_2 \neg X(T_2)$ we could conclude $M(X, T')$ with any time T' , whereas when we say *John might come Tuesday* this is certainly not implicationally equivalent with *John might come Wednesday*. Rather, *might* implies both agency and causal control, so that when John might come this means both that it is within his power to come and that unless he sets his mind on this it won't happen. This logic, being embedded in the lexical definition of the word *might*, is so strong that it extends even to cases where our contemporary thinking fails to see causal control, let alone agency and free will, to be at play. Consider the weather. When we say *The sun might shine* what this means is that the Sun, as an agent, can decide to come out from hiding behind the tree. The reference to the traditional children's song "Oh Mister Sun, Sun, Mister Golden Sun" may imply to some readers that the primitive animistic viewpoint whereby the Sun has the power to change its behavior is a vestigial remnant of a mode of thought restricted to kindergarten, yet the Wall Street Journal will use the exact same language about how stocks may rise or how the market can wipe out the gains it made in the past two weeks.

So far, we have a unary modal operator $M(X)$ that simply abbreviates the fact that some matter X *might* come about, a binary modal operator $M(X, T)$ that says it might come about at time T , and a ternary operator $M(A, X, T)$ that says that it might come about at time T by the agency of A . For the sake of completeness we could also add a binary operator $M(A, X)$ that says X might happen because of the agency of A but leaves the time unspecified. The standard approach would be to take the operator with the maximum arity as basic and define the others as special cases with some of the argument slots of the basic operator filled by some default value or quantified over. Here we

take the opposite tack, and argue that the basic operator has just one slot, for the matter X , and that the other slots are inherited from this simply because ‘matters’ in our sense can have agents, times, etc. But before getting into the details of this mechanism in Section 2, let us summarize what we have so far: *promise* is an ordinary verb whose agent X is also assumed, by default, to be the causal agent who brings about the promised matter X . The object X of the promise is typically expressed by an infinitival (as in *She promised to come*), a future tensed *that*-clause (as in *She promised that she will come*) or simply as some noun phrase or combination of noun phrases (as in *She promised complete immunity in return for a full confession*). The time of making the promise, T_0 is in the past relative to the time T that is relevant for the object of the promise, and from $P(A, X, T_0, T)$ we can conclude (\Rightarrow) both $C(A, X, T)$ and $\neg X(T_0)$.

Under the assumptions made here predicate arguments are handled quite differently from the way one would naively assign the participant roles. In the case of immunity, we assume the promisor p is in a position to cause some suspect s to have immunity against prosecution q for some misdeed d , and that it is s who needs to confess to d . Yet the sentence is perfectly compatible with a more loose assignment of roles, namely that the actual misdeed was committed by some kingpin k , and s is merely a witness to this, his greatest supposed crime being the withholding of evidence. This d' , being an accessory after the fact, is of course also a misdeed, but the only full-force implication from the lexical content of *immunity* is that there is some misdeed m that could trigger prosecution against which s needs immunity, not that $m = d$ or $m = d'$. The hypothesis $m = d$ is merely the most economical one on the part of the hearer (requiring a minimum amount of matters to keep track of) but one that can be defeated as soon as new evidence comes to light.

2 The tectogrammar of *promise*

Our method of analysis relies on unary (intransitive) predicates such as **promise**(X), **prosecute**(Y), **commit**(Z), **misdeed**(W), **immune**(V) and so forth, and on some lexical implications, expressed in terms of binary (transitive) predicates of what it means to DO or HAVE these things. (For now, we retain function/argument notation with variables to present these, but the formal system defined in Section 3 will not make use of variables.) Since to the mathematical logician the temptation to look at these as instances of Currying is almost irresistible, we want to make clear at the outset that in what follows the operation $A(B)$ ‘apply A to B ’ does not imply in any way that some intermediate function which takes functions as arguments was created. In fact, there is no implication that A or B are functions, and as we argue in Section 3, it is better to think of them as algebraic structures of a particular kind, *machines* (Eilenberg 1974). Yet somehow, with or without variables, the function-argument structure needs to be specified, which is precisely the task of tectogrammar (Curry 1961).

In order to deal with the external (subject) argument, we introduce an operator MAKE for which the external argument is obligatory. Taking the nominal

meaning of *promise* as basic, this means that *to promise* is derived from this nominal by application of the (morphologically implicit) MAKE: the expression *s promises X* will be analyzed as $\text{MAKE}(s, \text{promise}(X))$. The use of implicit operators has a long tradition, going back at least to generative semantics where the standard analysis of *kill* was ‘cause to die’. The use of unary operators is less widespread, and implies a significant departure from the standard mode of analysis whereby *She promised immunity for a confession* would be analyzed as *immunity* being the object of the promise, and *confession* as a free adverbial, outside the subcategorization frame of *promise*. The unary mode of analysis forces us assume that there is a single element, *immunity for a confession*, that is the object of the promise. What this means is that we must recognize another silent element, one that we will call **deal**, ‘something for something’, as an integral part of the analysis. This is confirmed by the communicative ease of introducing a definite description in a following sentence *The deal was rejected*. Further analysis of *deal* as ‘trade presented by the offeror as advantageous to the other party’ would be possible, but we do not pursue this here, since the main idea, that a promise has a single matter as its object, is already clear.

The same analysis is offered in regards to the time parameters, which are also standardly viewed as free adverbials. It is clear that the making of the promise has a temporal parameter. All finite verbs have an inflectional slot for this purpose, so this much is clear irrespective of one’s stance on using an implicit MAKE operator. This is the parameter we denoted by T_0 above. A consequence of our analysis is that if the object X has a time parameter T this is part of the promise, rather than being a free adverbial: if $\neg X(T)$ the promise is considered unfulfilled.

Again, the same analytic method can be applied to the causation predicate $C(A, X, T)$: instead of three direct arguments, we assume that the agent A is the subject of a head operator MAKE and the object X may, but need not, carry a temporal parameter of its own. There are many subtle issues concerning temporal causation, e.g. when by placing a bomb in Bob’s car on Monday Alice causes Bob to die on Tuesday, but we can largely skirt these as the central issue here is the promise, rather than the causal control required to keep it. It is worth keeping in mind that the typical failure mode of promises is not by failure to exert causal control but rather bad faith or forgetting: in most cases of broken promises the promisor could have done the right thing but didn’t, out of forgetfulness, or simply because the promise was not in earnest to begin with.

Finally, the same method works for M : there is a single argument, some matter X that might come about, but there is no time parameter other than the one that X may bring in, and for agentless cases there is no agent either. Thus *It might rain* is formulated $M(\text{rain})$ and *It might not rain* is formulated $M(\neg \text{rain})$. Based on the analysis offered so far, these two mean the same. However, if we consider the agentful cases, such as *John might insist on a vegetarian meal*, which is $M(\text{John insists})$ and *John might not insist on a vegetarian meal*, which is $M(\text{John} \neg \text{insists})$ the implications are very different: in the first case *we better tell the caterers* is reasonable, in the second *maybe we don’t have to bother the*

caterers is. Notice, however, that these implications concern our future plans not those of the agent: for the *might rain* case *we better set up a tent* is reasonable, for the *might not rain* case *maybe we don't have to set up a tent* is. What is really at stake are the plans of the hearer (irrespective of whether the act is by God or by John) to which we turn now.

In Rawls' words, promises are means to tie down the future. Simply put, $P(A, X, T_0, T)$ is *kept* by $X \Rightarrow X(T)$ or, by contraposition, it is reasonable to infer that the promise was not kept (or no promise was made) if we observe $\neg X(T)$. By the analysis presented above, both time and agent parameters can be eliminated from the argument structure: a promise X is kept if X , broken if $\neg X$. If Alice promises Carol twenty dollars, and Bob, a mutual friend, gives it to Carol the next day saying that it came from Alice, Carol will consider Alice's promise kept. If Bob just leaves the money on Carol's desk, Carol will not particularly know (or care) whether it came directly from Alice or not, she will likely assume that it did. However, if Carol finds the twenty dollar bill on the pavement she will not assume that Alice kept her promise. What this little example shows that the assumption of causation is still very much part of the meaning of promise. But if $P(A, X, T_0, T) \Rightarrow C(A, X, T)$ is now replaced by $P(X) \Rightarrow C(X)$, what means do we have to guarantee the identity of the promissor and the causer?

To answer this question we must invoke the external argument (Roeper 1987, Sichel 2009). Recall that the object of the promising, the matter X , is a promise because the promissor A *made* this promise. How did A make the promise? Obviously, she was doing things with words, she said *I promise*. It is evident that the agent of a performative is the performer, and the way to create a performative is by saying it. Rather than analyzing *s promises X* as $\text{MAKE}(s, \text{promise}(X))$ we will take into account the specific manner of making and analyze it as $\text{say}(s, \text{promise}(X))$ or better yet, $\text{say}(s, P)$ where the object of the saying happens to be a promise P . Notice that the exact same analysis is available for other performatives such as *deny* or *name* (as in *I name this ship Marie Celeste*): all that is required is to have a denial, or a name, as the object of saying.

Saying requires a recipient the same way causation requires an agent. It is possible that the default recipient is everyone, as in *proclaim*, or some higher power, as in *swear*, and in fact swearing (an oath) is meaningless without the assumption of such a higher power. But in the cases of central interest, communication between individuals for the purpose of making plans, promises are made to the hearer by the speaker, and the implication $P(X) \Rightarrow C(X)$ can be kept: the maker of the promise, the sayer, is the person held responsible for causing X to come about. Given our larger commitment to eliminate higher arity predicates, introducing a ditransitive $\text{say}(A, O, R)$ is a step of dubious utility. To simplify the analysis, we therefore take *say* to be analogous to *give* and analyze it as 'give words'. By giving a physical object X to R we create a situation where $\text{HAS}(R, X)$ will be true. By giving our word, we create a promise.

Adding the recipient to the picture, the analysis becomes *s promises X to R* meaning *s causes R to have s's word that X* or simply $\text{CAUSE}(R, \text{HAS}(s, \text{word}(X)))$.

It is not necessary for the promise to be addressed to the recipient, in fact a strong promise may explicitly invoke some higher recipient such as God. The real issue is how this giving of words, especially to beings whose very existence is doubtful, can nevertheless facilitate ‘tying down the future’. As Rawls argues, a promise is a promise to refrain from reevaluating later i.e. to go with the valuation at the time of the promising. When Alice says on Monday *Carol you will get your twenty bucks back before Friday* what this means that on Monday Alice values highly Carol’s having the money by Friday, and will do things to make this happen, such as going to the ATM and withdraw cash on Tuesday, or begging Bob to loan her a twenty on Wednesday so that she can pay Carol back.

Generative semanticists were largely content to use natural language paraphrases, saying *kill* means ‘cause to die’. Here we sketched a theory that is only slightly more formal, saying x *kill* y means ‘ x CAUSE die(y)’. By introducing explicit role variables, and typographically encoding the distinction between unary and binary predicates, the notation is more capable of exposing the tectogrammar than reliance on the infinitival *to*. This actually neutralizes a central point of Fodor’s (1970) critique of the generative semantics analysis, because arguments concerning the placement of pronouns are no longer applicable. (As a matter of fact, subsequent developments in binding theory also rendered this kind of criticism irrelevant.)

The key reason for using *to* in the paraphrase was the commitment that generative semantics had to utilizing phrase-markers (context-free trees) as underlying structures, and the assumption that deep structure is the appropriate place to fix the lexical category of the words (Lakoff 1968). It is clear from the foregoing that we are quite content treating *promise* as entirely neutral between nominal and verbal, and forming the verbal version by zero affixation of MAKE. This is one point where the work presented here departs quite strikingly from the generative semantics tradition, reaching back straight to Pāṇini, who also was a generative semanticist in the sense of deriving surface form from underlying meaning, but was also more of a morphologist, deriving both nominal and verbal forms from the same root.

Fodor’s final argument is based on on the perceived arbitrariness of the decomposition: why stop at ‘cause to die’, why not go to ‘cause not to live’ or ‘cause not to have life functions’ and so on? This criticism is pertinent not just to generative semantics, but in fact to any system where the meaning of one entity is described in terms of other entities. There are two known ways out: first, designating a fixed set of primitives where decomposition stops. This is the approach taken both by the Longman Dictionary of Contemporary English, where a set of about two thousand primitives is used (Boguraev and Briscoe 1989), and by the NSM school (Wierzbicka 1985). The second way out is to use an algebraic, rather than logic-based, theory of decomposition (Kornai 2010a), which is immune to the charge of arbitrariness of primitives the same way linear spaces are independent of the choice of basis we use to present them: the choice is arbitrary, but one choice is just as good as the other.

3 The formal model

For Russell, whose chief interest was with providing logical foundations for mathematics and the sciences, the Aristotelian maxim of Leibniz that predicates are inherent in their subject was completely untenable, since such an assumption would make it impossible to handle asymmetric cases like the predicate *father*. The differences between *Mick fathered Nixon* and *Nixon fathered Mick* are easily seen in the implications (defaults) associated with the superordinate (parent) and subordinate (child) slots: the former is assumed to be independent of the latter (already existed before the act of fathering took place), the latter is assumed to be dependent on the former, the former controls the latter (in the same everyday sense of control that we used so far, not in the grammatical sense), and not the other way around, etc.

In our treatment of verbs, it will indeed be necessary to admit at least one asymmetric relation, which we will denote ‘<’, and perhaps a handful of others such as HAS(x,y) ‘x possesses y’ or AT(x,y) ‘x is at location y’. At the same time, we are more parsimonious with relations than Russell, for whom the existence of a single asymmetrical relation was sufficient reason to open the floodgates and admit all kinds of relations, and presented a theory in which no ternary relations are used in the definiens. We illustrated our method of analysis on a hard case, *promise*, that is standardly thought to require at least three, and possibly as many as five, arguments, and argued that at the tectogrammatical level it has only *one* argument, the thing that is being promised. All other arguments are linked in either externally (the promissor, by the matrix verb MAKE) or recursively, by invoking the frame of the act of promise-making (which we analyzed as an act of giving words), or the frame of the matter being promised.

To round out this picture what we need is a theory of the representational objects, one that describes how semantic representations are formed, maintained, and destroyed (see 3.1) and a theory of bookkeeping that tells us how such objects can act as slot-fillers in the tectogrammar (see 3.2). (Ideally, we would also want an account of the phenogrammar, how all these steps are realized on the surface, but this is clearly beyond the scope of this paper.)

3.1 Representation by machines

Fortunately, a good theory of representational objects is already at hand: these are the *machines* of Eilenberg (1974). In brief, *a machine is a mapping between the alphabet of some FSA and the relation monoid of some set X*. Eilenberg intended machines to be an algebraic formulation of the flowcharts widely used at the time for describing the structure of computer programs – we will use them to represent the meaning of morphemes, words, phrases, sentences, and texts alike. The FSA is used as the *control* of the device just as in Turing Machines, and the relations are best thought of as transformations of the base set X that the machine is about.

Definition 1 A *machine* with an alphabet Σ over a base set X is given by an *input set* Y ; an *output set* Z ; a relation $\alpha : Y \rightarrow X$ called the *input code*; a relation

$\omega : X \rightarrow Z$ called the *output code*; a finite state automaton $\langle S, T, I, F \rangle$ over Σ called the *control FSA*; and a mapping M of each $\sigma \in \Sigma$ to some $\phi \in \Phi \leq 2^{X \times X}$.

Since our objects are semantic representations for natural language expressions rather than flowcharts, we need to tweak this definition a bit. As we are not dealing with the phenogrammar, we can safely ignore the input and output mappings, which are primarily formal tools for transducing input to, and output from, the machine. This will simplify the definition, but we also need to complicate it a bit: we need to be more specific about the base set X , whose elements will be called *partitions*, and we will need to designate one of these partitions as the *head*. One partition (conventionally numbered as the 0th member of the set X) will contain the phonological form (printname) of the machine, the other(s) will store information relating to the argument(s).

We will call the machines so defined *lexemes*, and informally it is best to think of these as monolingual dictionary entries (see Kornai 2010). One characteristic difference between the model-theoretic and the more cognitively inspired theories of lexical semantics is the type structure: Montague Grammar relies on a strict set of intensional and extensional types, with n -ary predicates and relations, while lexical semantics is generally conceived of in network terms, with only two main types, graph *nodes* corresponding to lexemes, and graph *edges* corresponding to various links, directed or undirected. From the perspective of strict typing, it is natural to ask how property bundles are composed: for example, if properties correspond to qualia, is it simply the case that adjectives are qualia and nouns are bundles of qualia? From the perspective of the essentially type-free network theory, the main question is to sort out the kinds of links permitted by the model (Woods 1975). Here we will try to sketch an answer to both kinds of questions.

Primitive lexemes come in two subvarieties, *unary* and *binary*: the classes will be denoted by U and B and the instances written in **typewriter font** and SMALL CAPS respectively. Most lexical entries, not just nouns, adjectives, and intransitive verbs, but also verbs of higher arity (transitives, ditransitives, etc.), both in predicative and in substantive forms, are viewed as unary, and the binary category is reserved primarily for adpositions (both pre- and postpositions) and case markers. With adpositions, it is very hard to see how expressions signifying pure spatial relations such as *under* or *near* could be given a satisfying model without reference to the pairs of objects standing in the named relation, and from a grammatical perspective it is quite clear that case markers behave very similarly (for a modern summary, see Anderson 2006). There are a few stray examples elsewhere in the system of grammatical formatives, such as the possessive relation, generally not regarded a true case, and the comparative morpheme *-er*, but it is clear that on the whole binary lexemes are restricted to a small, closed subset of function words, while the large, productive classes of content words are all unary under the analysis offered here.

Definition 2 The surface syntax of lexemes can be summarized in a Context-Free Grammar (V, Σ, R, S) as follows. The nonterminals V are the start symbol S ; the binary relation symbols B which can include '<', CAUSE, HAS, ... etc.

taken from some small fixed inventory of deep cases, thematic roles, grammatical functions, or similarly conceived linkers; and the unary relation symbols collected in U . Variables ranging over V will be taken from the end of the Latin alphabet, v, w, x, y, z . The terminals are the grouping brackets '[' and ']', the derivation history parentheses '(' and ')', and we introduce a special terminating operator ';' to form a terminal v ; from any nonterminal v . The rule $S \rightarrow U|B|\lambda$ handles the decision to use unary or binary predicates, or perhaps none at all. The operation of *attribution* is captured in the rule schema $w \rightarrow w; [S^*]$ which produces the list defining w . (This requires the CFG to be *extended* in the usual sense that regular expressions are permitted on the right hand side, so the rule really means $w \rightarrow w; []|w; [S]|w; [SS]|\dots$) Finally, the operation of *predication* is handled by $u \rightarrow u; (S)$ for unary, and $v \rightarrow Sv; S$ for binary nonterminals.

Our interest is both with the terminal yield of the grammar (V, Σ, R, S) and the sentential forms that still contain nonterminals. The meaning postulates are specific instances of the attributive rule schema $w \rightarrow w; [S^*]$ which produces the list defining w and the predicative schemas $u \rightarrow u; (S)$ and $v \rightarrow Sv; S$. Whenever such a postulate is used, the definiendum x is terminated (replaced by the terminal x ; and thus no longer available for further rewriting), but the substantive terms that occur in the definiens are still in nonterminal form. Before drawing many conclusions from the fact that the syntax is defined as context-free it is worth emphasizing that this is *pure* syntax. Thus, **dog EQ four-legged, animal, hairy, barks, bites, faithful, inferior** is a well-formed equational formula defining the dog, but so is **cat EQ barks** – the syntax is entirely neutral as to whether this is true or what sense it makes. The standard method of trying to make sense of such formulas would be to interpret them in model structures, and failure to do so is generally seen as failure of connecting language to reality (Lewis 1970, Andrews 2003). Yet, as we have argued elsewhere (Kornai 2010b), such an effort is bound to misfire wherever we encounter language that is not about reality.

Consider *Pappus tried to square the circle/trisect the angle/swallow a melon*. In one case, we see Pappus intently studying the works of Hippocrates, in the other we see him studying Apollonius, and in the third case we see him in the vegetable patch desperately looking for an undersized melon in preparation for the task – clearly the truth conditions are quite different. We may very well imagine a possible world where throats are wider or melons are smaller, but we know it for a fact that squaring the circle and trisecting the angle are logically impossible tasks. Yet to search for a proof, be it positive or negative, is quite feasible, and the two searches lead us into different directions early on: squaring the circle begins with the Hippocratic lunes, and culminates in Lindemann's 1882 proof, while trisecting the angle begins with the *Conics* of Apollonius and does not terminate until Wantzel's 1832 proof. The problem is not with nonexistent objects such as superwide throats, for which the intensional treatment of opacity works fine, but also *necessarily* nonexistent objects whose extension is empty at every index. (To make matters worse, we rarely know in advance whether something fails to exist by accident or of necessity.)

In truth, it is not just the existence of hard hyperintensionals that stands in the way of ever completing the program of model-theoretic semantics – the failure of this approach is more evident from ordinary sentences than from subtle technical notions concerning hyperintensionals, which may yet get resolved by work such as Pollard (2008). Consider, for example, the following statement, (Jonathan Raban, NYRB 04/12/07): *There is in Sullivan’s makeup [] an Oxford debater’s ready access to the rhetoric of condescending scorn*. Clearly, this is a completely meaningful, non-paradoxical sentence, which conveys good information about Sullivan to the readers of the New York Review, yet attempts to analyze it in terms of satisfaction in model structures are fruitless. It is quite unclear who is, and who isn’t, an Oxford debater, or how we could go about distinguishing an Oxford from a Harvard debater in terms of the set of people involved (especially as most debaters are perfectly capable of switching between the various styles of debate). The same can be asked about every constituent of the sentence: where is, in a model structure, *someone’s makeup*, and what kind of objects *r* are we sifting through to determine whether *r* is or is not part of Sullivan’s makeup? What is *scorn*, and are Lewis’ (1970) remarks on Markereese really exemplars of the *condescending* variety, or are they, perhaps, well reasoned and not at all scornful?

The semantics that attaches to the lexeme-based representations defined above by purely syntactic means is of a different kind. We may not have a full understanding of the relation *x has ready access to y*, but we do know that having ready access to something means that the possessor can deploy it swiftly and with little effort. What the sentence means is simply that Raban has studied the writings of Sullivan and found him capable of doing so, in fact as capable as those highly skilled in the style of debate practiced at the Oxford Union where condescension and scorn are approved, even appreciated, rhetorical tools. It is basically left to the reader to supply their own understanding of *condescension* and *scorn*, and there is no reason to believe that this understanding is framed in terms of specifying at every index whether something is condescending or scornful. Rather, these terms are either primitives, or again defined by meaning postulates.

A defining characteristic of this network of definitions is that little semantic distinction can be made between verbs like *to promise*, *to prosecute*, *to commit*, *to (be/make) immune*, *to *misdo*, their substantive forms *promising*, *prosecuting/prosecution*, *commitment*, *immunity*, **misdoing*, and their cognate objects *the promise*, *the prosecution*, *the commitment*, *the misdeed*. In this respect, the underlying type system proposed here is considerably less strict than that of Lakoff (1968), where deep structure was assumed to be the appropriate place for fixing the lexical categories of the words. But this kind of loose typing, the necessity of which is a central claim in Turner (1983, 1985), is quite suitable for a purely lexical theory, like that of Pāṇini, which can capture the essential grammatical parallelism between active, passive, and stative constructions (see Kiparsky 2002:2.2). We also stay close to the Pāṇinian model in assuming that the argument structure, such as it is, is created by the linkers. To illustrate

the mechanism, consider *give*(x, y, z), which is standardly analyzed as ‘transferring possession of y from x to z ’. From our perspective, such an analysis is assuming too much, because when we say *The idea gave him the shivers* one cannot reasonably conclude that the shivers were originally in the idea’s possession, and when we say *Mary gave him typhoid*, we cannot conclude that Mary ceased to have typhoid just by giving it to him. Thus we have a simpler analysis, $\text{CAUSE}(x, \text{HAS}(z, y))$ ‘cause to have’ where CAUSE is used to denote the agentive linker.

It is worth noting that the formalism offered above does not rely on function/argument notation and variables at all. To do away with these entirely, we already fixed the notation: since the binary operators can be written infix, while unary operators are written prefix, parens are sufficient to fix the location (though not the identity) of the variables: a formula such as $x \text{ CAUSE}(z \text{ HAS } y)$, can be reduced to $\text{CAUSE}(\text{HAS})$. The example is only illustrative of the formal mechanism – this is not the place to recapitulate the subtleties of causation discussed in Talmy (1988), Jackendoff (1990:72) and elsewhere in the linguistic literature. By assuming right association most parens can be omitted, only those signaling left association need be retained to disambiguate application order if necessary (so far we have not found actual examples). For grouping, braces will be used, so that the conjunctive feature bundles defining nouns can be kept together. Such a tight notation does not leave a great deal of room for scope ambiguities, but as we have argued in some detail elsewhere (Kornai 2010a), this entails little loss in that universally quantified expressions, outside the technical language of mathematics, are read generically rather than episodically.

Eliminating variables is a significant step toward bringing the formalism closer to the network diagram notation familiar from many works in lexical semantics and Knowledge Representation (for a good selection, see Findler 1979, Brachman and Levesque 1985). We cannot discuss the network aspect of the theory here in sufficient detail, but we note that in the machine formalism the proliferation of links, characteristic of many network theories, is kept under strict control. This is achieved by two means: first, IS_A links are derived rather than primitive (see Kornai 2010), and second, by the elimination of ditransitives.

Were we to permit ditransitives and higher arity predicates as primitives, we would need as many kinds of links as the maximum arity predicate has arguments, and to the extent this number is treated as an unlimited resource (as in some analyses of serial verbs) we would need to countenance an infinite number of link types. As it is, we are restricting the theory to only two kinds of links: those corresponding to substitution of the first argument, and those corresponding to the substitution of the second (as a matter of fact, ergative/absolutive classification of links would be just as feasible, but we do not pursue this alternative here).

3.2 Slot-filling

The only fundamental aspect of the theory not discussed so far is the book-keeping, how to specify which empty slot in a machine corresponds to which

verbal argument, how to guarantee that no slot gets filled twice, and in case of obligatory arguments, how to guarantee that the slot does get filled. Recall that Definition 1 contains two moving parts, an FSA and a base set X , as well as a mapping from the alphabet of the automaton to the set of relations over X . This, we claim, is already sufficient for the purposes of tectogrammar. Unaries, by their very nature, have only one slot to be filled, so linking something there requires no traffic signals: wherever X is an unary and Y is an arbitrary machine $X(Y)$ is obtained by placing an instance of Y on the one and only non-phonological partition of X .

For the binary case, consider *Mick fathered Nixon* and assume that *father* is a relational noun or that *to father* is a transitive verb. What we wish to obtain (using infix notation) is **Mick FATHER Nixon** rather than **Mixon FATHER Mick** or **Mixon, Mick FATHER** or something else. We will ignore the tense marking, and we will assume a rather sophisticated phenogrammar that has already succeeded in turning the surface expression into **Mick-NOM, Nixon-ACC, FATHER**. In English, the nominative and accusative linking is provided by word order, in other languages it may very well be provided by overt case marking. (In fact, it is slightly wrong to use the terms nominative and accusative in that the two slots may as well be linked by ergative and absolutive case, but this affects only the phenogrammar of the language in question, not the mechanism proposed here.)

It is sufficient for the alphabet of the control automaton of the FATHER machine to distinguish three elements, those NPs that are nominatively marked, for which we use the letter n , those accusatively marked, for which we use the letter a , and all others, denoted by o (see Fig 1). Since *to father* is transitive, the control FSA will be a square, with a start state we denote by \odot , an accepting state \bullet , and two other states serving as counters for unfilled valences. The language accepted by the automaton is the shuffle product of exactly one a , exactly one n , and an arbitrary number of os .

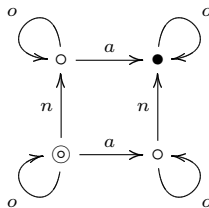


Figure 1 FSA for transitive verbs

The control is used to define a mini-language that checks the tectogrammatic conditions: for example for verbs that alternate in transitivity such as *eat* the top left state could also be defined accepting, so that *Mick ate*, unlike **Mick fathered*, would come out as grammatical.

The mapping M is also part of the bookkeeping mechanism. Continuing with the example of FATHER, let us denote the two partitions 1 and 2. The relations possible over these include $F = \{(1, 1), (1, 2), (2, 1), (2, 2)\}$; $I = \{(1, 1), (2, 2)\}$; $P =$

$\{(2,2)\}$ and $Q = \{(1,1), \}$ (there are a total of 16 relations over two elements, but the others need not concern us here). Here we map by M the letter o on the identity relation I , the letter a on the projection P and the letter n on the projection Q . As we build up a string, we are also building up a product of relations, so from starting the full relation F , by the time we multiplied with exactly one P , one Q , and any number of I s, we arrive at the empty relation. The mechanism is flexible enough to handle complex relation-changing verbal affixation rules such as passivization or causativization.

Finally, let us consider how the ‘cause to have’ analysis of *give* is formalized using machines. The square FSA of Fig. 1 is replaced by a cube, whose edges are now labeled n (ominative), d (ative), a (ccusative), and o (ther), though the loops labeled o that appear over each vertex are omitted from the figure for clarity.

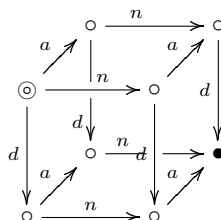


Figure 2 FSA for *give*

Assuming all three arguments are obligatory, there is only one accepting state, the bottom back right corner of the cube. The base set X has three members (not counting the phonological partition), which are obtained by substituting the HAS machine in the second (subordinate) partition of the CAUSE machine. In a network diagram, this is depicted as Fig 3 below, with nodes both for binary and unary machines, and different coloring (straight vs. dotted) of the edges to make clear which edge originates in the first, and which in the second partition of the binaries.



Figure 3 Base set for *give*(x,y,z)

4 Conclusions

Since grammars need to capture tectogrammatical generalizations, some form of slot-filling mechanism, such as the f -structure of LFG or the SUBCAT mechanism of HPSG, is clearly needed for dealing with predicate-argument structure. Indeed, the need is felt so strongly that a variety of linguistic theories such as case grammar (Anderson 2006), valency theory (Somers 1987) and tagmemics (Pike

1960) posited slot filling as the basic (and in some cases, the only) mechanism for describing syntactic phenomena.

From a formal standpoint the most immediate mechanism for slot-filling is to use some kind of variable binding term operators, typically lambdas, as in $\lambda x \lambda y \lambda z \text{ give}(x, y, z)$. Once we take this step, the elimination of ditransitives, and indeed the elimination of transitives, becomes a trivial matter of currying, and attention is shifted to other aspects of the system: as is well known (Marsh and Partee 1984), variable binding itself is a formally complex operation, with attendant difficulties for creating effective parsing/generation/acquisition algorithms.

In the machine formalism propounded here it would actually be possible to have ditransitives or even higher arity predicates, but only at a computational cost that increases superexponentially. For technical reasons n -ary predicates require machines with base set cardinality $|X| = n + 1$ (the 0th slot is used for storing the phonological, morphological, and other position-independent information) so the number of distinct binary relations ϕ is $2^9 = 512$, the number of ternaries would be $2^{16} = 65,536$, the number of quaternaries $2^{25} = 33,554,432$ and so on.

Note that the empirical distribution of higher arity verbs drops off rather sharply: in English we have tens of thousands of intransitive and transitive verbs, but only a few hundred ditransitives, and only a handful of candidates for tritransitive or higher arity. Following Schank (1973), the single most frequent class is physical transfer (PTRANS) verbs such as *give*, *get*, *bring* and negative PTRANS such as *bar*, *block*, *keep* – altogether less than thirty examples including portmanteau manner-qualified forms such as *throw*, *toss* and *mail* where the indirect object is arguably optional. The next most frequent class is mental transfer (MTRANS) verbs like *signal*, *promise*, *inform*, *show* followed by transfer of possession (ATRANS) verbs such as *award*, *bequeath*, *remit* and their negatives such as *begrudge*, *deny*, or *refuse*. The M and A classes already show signs of morphological complexity, and in languages that have overt causative or benefactive morphology the higher arity classes are somewhat larger, but still a small fraction in terms of token frequency.

This faster than exponential frequency dropoff is hard to grasp from the variable-binding standpoint, where currying is always available, but makes perfect sense from the machine standpoint, where creating (acquiring) and operating (during parsing and generation) larger machines would require disproportionately larger resources. In this regard, the current work fits far better with variable-free (Szabolcsi 1987, Jacobson 1999, Steedman 2001) than with mainstream semantics. However, the fit is far from perfect, in that machines are best thought of as a means of capturing the structure of meaning postulates, rather than as a calculus for compositional meaning. Of the two, we actually consider lexical (non-compositional) structure the higher priority task, given that the primary information source in a sentence, responsible for over 85% of the information conveyed, is the choice of the words, rather than the grammatical structure,

which accounts of less than 15% (see Kornai 2010 for how these numbers are obtained).

Altogether, the theory presented here fits better with the ‘cognitive’ approach pursued by Jackendoff, Talmy, Langacker, Fauconnier, Lakoff, Wierzbicka, and many others, and with the whole network tradition of Knowledge Representation originating with Quillian (1967) and Schank (1973). One issue that has put the cognitive work on a less than equal footing with the Montague Grammar tradition was the naive formalism (famously dubbed ‘markerese’ by Lewis 1970), and part of our goal is to provide a formal apparatus that is capable of restating the linguistic insights of the cognitive work in a theory that is sufficiently formal for computer implementation.

Readers familiar with the history of network theories will know that one of the key implementational issues is the variety of links permitted in the system (see in particular Woods 1975), and in this regard the elimination of ditransitives is a key step. In a network graph, every edge from a node x to some node y and bearing the label l is of necessity an ordered triple (l, a, b) i.e. an information structure with three slots. A theory that makes the claim that these are not unanalyzed primitives but can be built from simpler, binary structures enables reduction of complexity across the whole system. Specifically, we claim that there are only *two* kinds of links (depicted by full vs. dotted lines in Fig.3), corresponding to the superordinate (first) and the subordinate (second) slot of binary relations. There is no claim that first always means ‘1’ or subject, and second means ‘2’ or object, the formal theory presented here is quite capable of handling mismatches such as experiencer subjects. The claim is simply that there is never a ‘3’ or indirect object on a par with the first two arguments.

To summarize, we have repurposed Eilenberg’s machines as a simple, variable-free mechanism for decomposing the meaning of higher arity relations and keeping track of the tectogrammar (function-argument structure). This is the hard case: extending the system to adjectival and adverbial modifiers is trivial and requires no further machinery (see Kornai 2010). The result is a formalism conducive to the style of grammatical analysis familiar from Pāṇini and from generative semantics, and capable of encoding the semantic insights developed from Aristotle to contemporary knowledge representation and cognitive semantics.

Acknowledgments

We thank Donca Steriade (MIT) for comments on an earlier draft. Work supported by OTKA grants #77476 (Algebra and algorithms) and #82333 (Semantic language technologies).

References

- Anderson, J. (2006). *Modern grammars of case: a retrospective*. Oxford University Press

- Andrews, A. (2003). Model-theoretic semantics as structural semantics. *ms, ANU*.
- Boguraev, B. K. and Briscoe, E. J. (1989). *Computational Lexicography for Natural Language Processing*. Longman.
- Brachman, R. (1979). On the epistemological status of semantic networks.
- Brachman, R. and Levesque, H. (1985). *Readings in knowledge representation*. Kaufman Publishers Inc., Los Altos, CA.
- Curry, H. B. (1961). Some logical aspects of grammatical structure. In Jakobson, R., editor, *Structure of Language and its Mathematical Aspects*, pages 56–68. American Mathematical Society, Providence, RI.
- Eilenberg, S. (1974). *Automata, Languages, and Machines*, volume A. Academic Press.
- Findler, N (1979). *Associative Networks: Representation and Use of Knowledge by Computers*. Academic Press.
- Fodor, J. (1970). Three reasons for not deriving “kill” from “cause to die”. *Linguistic Inquiry*, 1(4):429–438.
- Graham, AC (1958). *Two Chinese Philosophers*. London.
- Jackendoff, R. S. (1990). *Semantic Structures*. MIT Press.
- Jacobson, P. (1999). Towards a variable-free semantics. *Linguistics and Philosophy*, 22:117–184.
- Kiparsky, P. (2002). *On the Architecture of Pāṇini’s grammar*. *ms*, Stanford University.
- Kornai, A. (2008). *Mathematical Linguistics*. Springer Verlag.
- Kornai, A. (2010a). The algebra of lexical semantics. In Jäger, G. and Michaelis, J., editors, *Proceedings of the 11th Mathematics of Language Workshop*, FoLLI Lecture Notes in Artificial Intelligence. Springer Verlag.
- Kornai, A. (2010b). The treatment of ordinary quantification in English proper. *Hungarian Review of Philosophy*, 54(4):150–162.
- Lakoff, G. (1968). Pronouns and reference.
- Lewis, D. (1970). General semantics. *Synthese*, 22(1):18–67.
- Marsh, W. and Partee, B. (1984). How non-context-free is variable binding? M. Cobler and S. MacKaye and M. Wescoat (eds) *Proceedings of the West Coast Conference on Formal Linguistics III*, 179–190.
- Pike, K. (1960). *Language in Relation to a Unified Theory of the Structure of Human Behavior*. Mouton, The Hague.
- Pollard, C. (2008). Hyperintensions. *Journal of Logic and Computation*, 18(2):257–282.
- Quillian, M. R. (1969). The teachable language comprehender. *Communications of the ACM*, 12:459–476.
- Rawls, J. (1955). Two concepts of rules. *The Philosophical Review*, 64(1):3–32.
- Roeper, T. (1987). Implicit arguments and the head-complement relation. *Linguistic Inquiry*, 18:267–310.
- Russell, B. (1900). *The Philosophy of Leibniz*. Allen und Andwin.
- Schank, R. (1973). *The Fourteen Primitive Actions and Their Inferences*. Stanford AI Lab Memo 183.

- Sichel, I. (2009). New evidence for the structural realization of the implicit external argument in nominalizations. *Linguistic Inquiry*, 40(4):712–723.
- Somers HL (1987). *Valency and case in computational linguistics*. Edinburgh University Press.
- Sowa, J. (2000). *Knowledge representation: logical, philosophical, and computational foundations*, volume 594. MIT Press.
- Steedman, M. (2001). *The Syntactic Process*. MIT Press.
- Szabolcsi, A. (1987). Bound variables in syntax – are there any? In J. Groenendijk, M. Stokhof, and F. Veltman, editors, *Proceedings of the 6th Amsterdam Colloquium*, pages 331–351, Amsterdam. Institute for Language, Logic, and Information.
- Talmy, L. (1988). Force dynamics in language and cognition. *Cognitive science*, 12(1):49–100.
- Turner, R. (1983). Montague semantics, nominalisations and Scott’s domains. *Linguistics and Philosophy*, 6:259–288.
- Turner, R. (1985). Three theories of nominalized predicates. *Studia Logica*, 44(2):165–186.
- Wierzbicka, A. (1985). *Lexicography and conceptual analysis*. Karoma, Ann Arbor.
- Woods, W. A. (1975). What’s in a link: Foundations for semantic networks. *Representation and Understanding: Studies in Cognitive Science*, pages 35–82.